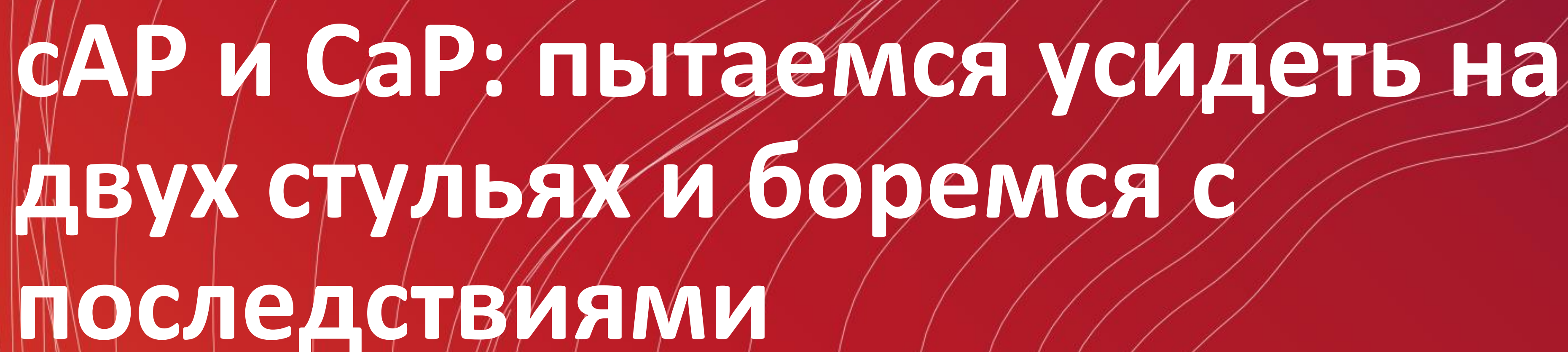


cAP и CaP: пытаемся усидеть на двух стульях и боремся с последствиями

Сергей Петренко



О себе

Меня зовут Сергей Петренко

О себе

Меня зовут Сергей Петренко

Занимаюсь репликацией в Tarantool



О себе

Меня зовут Сергей Петренко

Занимаюсь репликацией в Tarantool

Занимался алгоритмом синхронной репликации
и выборов лидера



План

- Введение
- Особенности Raft в Tarantool
- Проблема асинхронных транзакций
- Размещение кластера
  Raft в двух ЦОДах
- Обнаружение различных историй лидерства
- Выводы





CAP теорема

В присутствии проблем сети (P) система может гарантировать либо консистентность (C), либо доступность (A), но не то и другое вместе



CAP теорема

В присутствии проблем сети (P) система
может гарантировать либо консистентность (C),
либо доступность (A), но не то и другое вместе

- Консистентность – запрос на чтение возвращает все записи, подтверждённые до этого запроса



CAP теорема

В присутствии проблем сети (P) система
может гарантировать либо консистентность (C),
либо доступность (A), но не то и другое вместе

- Консистентность – запрос на чтение возвращает все записи, подтверждённые до этого запроса

- Доступность – запрос к любому из работающих узлов возвращает ответ за конечное время





Почему бы не выбрать CAP?

"foo" = 3

ack

read "foo"

nil





Виды репликации

Асинхронная



Синхронная



Виды репликации

Асинхронная



- Available

Синхронная





Виды репликации

Асинхронная



- Available

Синхронная



- Не available



Виды репликации

Асинхронная

- Available





- Не available
- Может быть consistent



Raft

- CaP
- кворум – большинство



План

- Введение
- Особенности Raft в Tarantool
- Проблема асинхронных транзакций
- Размещение кластера Raft в двух ЦОДах
- Обнаружение различных историй лидерства
- Выводы





В Raft нам не хватает режима cAP (вместо CaP)





Особенности нашей реализации Raft

- Выбор между синхронной и асинхронной репликацией



Особенности нашей реализации Raft

- Выбор между синхронной и асинхронной репликацией
- Настраиваемый кворум



План

- Введение
- Особенности Raft в Tarantool
- Проблема асинхронных транзакций
- Размещение кластера Raft в двух ЦОДах
- Обнаружение различных историй лидерства
- Выводы





async "foo" = 1

ack





async "foo" = 2

ack

async "foo" = 3

ack





"foo" = 1



"foo" = 2



"foo" = 3



"foo" = 2

"foo" = 3



А как же CheckQuorum?





План

- Введение
- Особенности Raft в Tarantool
- Проблема асинхронных транзакций
- Размещение кластера Raft в двух ЦОДах
- Обнаружение различных историй лидерства
- Выводы





Варианты размещения кластера в двух ЦОДах

- Поровну





Варианты размещения кластера в двух ЦОДах

- Поровну
- Дополнительный ЦОД для голосования



Снижение кворума при аварии

ЦОД 1

1

2

ЦОД 2

3

4



Снижение кворума при аварии

ЦОД 1

ЦОД 2

1

2

3

4

Tarantool



Снижение кворума при аварии

ЦОД 1

1

2

ЦОД 2

3

4

Tarantool



При ручном вмешательстве никто не застрахован от ошибок,
а кластер – от сплитбрейна





Ошибочное снижение кворума

ЦОД 1

ЦОД 2

promote

1

2

3

4

promote

Ошибочное снижение кворума

ЦОД 1

ЦОД 2

"foo" = 1

1

2

3

4

"foo" = 2

Tarantool



Ошибочное снижение кворума

ЦОД 1

ЦОД 2

1

2

3

4

"foo" = 1

"foo" = 2



Ошибочное снижение кворума

ЦОД 1

1

2

"foo" = 1

?!

"foo" = 2

ЦОД 2

3

4



Человеческие ошибки случаются и без аварий



Кворум может быть неправильно сконфигурирован

Человеческие ошибки случаются и без аварий



Человеческие ошибки случаются и без аварий



Кворум может быть неправильно сконфигурирован



Может появиться два лидера



Нужно справляться с последствиями работы двух лидеров и для синхронных транзакций



План

- Введение
- Особенности Raft в Tarantool
- Проблема асинхронных транзакций
- Размещение кластера Raft в двух ЦОДах
- Обнаружение различных историй лидерства
- Выводы





Репликация журнала

Node 1

Node 2

Node 3

t, время





Репликация журнала

Node 1

Node 2

Node 3

t, время

Репликация журнала

Node 1

Node 2

Node 3

разный порядок

t, время

Репликация журнала с Raft

Node 1

Node 2

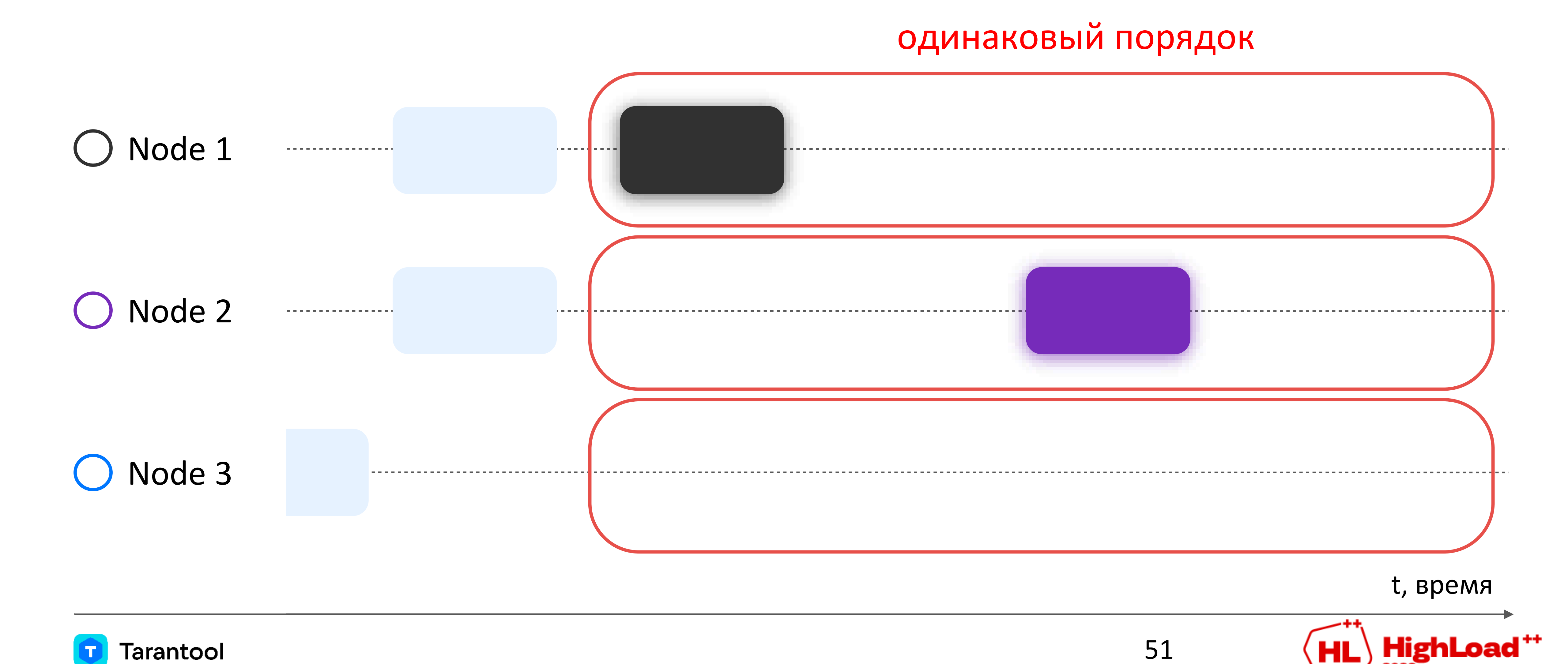
Node 3

t, время





Репликация журнала с Raft



Репликация журнала с Raft. Split-brain

Node 1

Node 2

Node 3

t, время





Репликация журнала с Raft. Split-brain

Node 1

Node 2

Node 3

t, время

Репликация журнала с Raft. Split-brain

Node 1

Node 2

Node 3

разный порядок

t, время

Владение журналом



Когда всё хорошо, PROMOTE разных узлов выстраиваются
в линейную историю





Записи в журнал при изменении лидера

Node 1 — t 5 ex

Node 2 — t 5 ex

Node 3 — t 5 ex

t, время

Записи в журнал при изменении лидера

Node 1

Node 2

Node 3

t 5
ex

t 6
ex 3

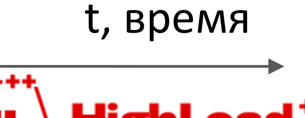
t 5
ex

t 999
ex 3

Видит, что пропала нода 3
Не видит, что избралась нода 1

t 5
ex

t, время



Старый лидер подтвердил транзакцию

Node 1

LSN =

LSN = 50

LSN = 51

Продолжает работать

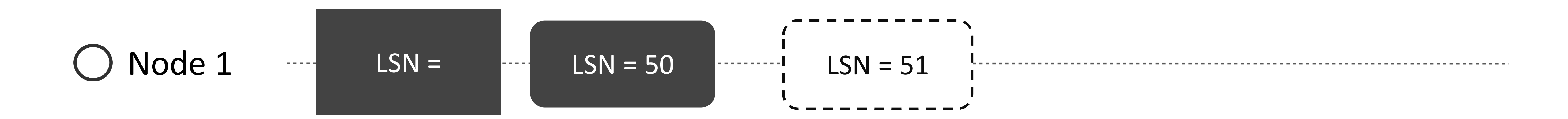
Node 2

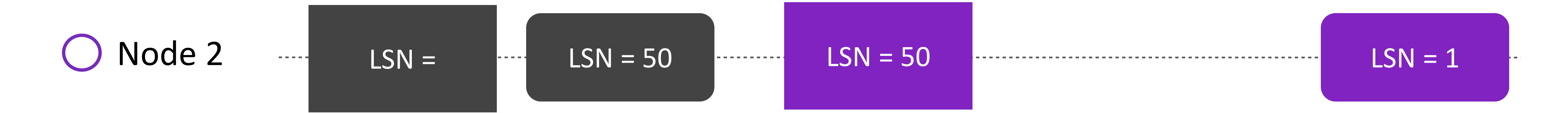
LSN = 50

Думает, что 1 нода умерла

t, время

Старый лидер подтвердил транзакцию





t, время





Почему мы не разрешаем конфликты автоматически?





План

- Введение
- Особенности Raft в Tarantool
- Проблема асинхронных транзакций
- Размещение кластера Raft в двух ЦОДах
- Обнаружение различных историй лидерства
- Выводы





Выводы

При наличии асинхронных транзакций даже Raft не защищает от перехода в master - master



Выводы

При наличии асинхронных транзакций даже Raft не защищает от перехода в master - master

Неправильная конфигурация вообще позволяет получить master-master даже с синхронной репликацией



Выводы

При наличии асинхронных транзакций даже Raft не защищает от перехода в master - master

Неправильная конфигурация вообще позволяет получить master-master даже с синхронной репликацией

Данные нужно защищать от последствий независимого изменения

Выводы

При наличии асинхронных транзакций даже Raft не защищает от перехода в master - master

Неправильная конфигурация вообще позволяет получить master-master даже с синхронной репликацией

Данные нужно защищать от последствий независимого изменения

Думаем о том, как упростить возвращение убежавшего лидера в кластер



The end

- При наличии асинхронных транзакций даже Raft не защищает от перехода в master - master
- Неправильная конфигурация вообще позволяет получить master-master
  даже с синхронной репликацией
- Данные нужно защищать от последствий независимого изменения
- Думаем о том, как упростить возвращение убежавшего лидера в кластер



